UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/919,743	07/31/2001	Stepan Sokolov	SUN1P832/P6211	3944

22434 7590 12/02/2004

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

EXAMINER
TANG, KUO LIANG J

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 12/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/919,743	**Applicant(s)** SOKOLOV ET AL.	
	Examiner Kuo-Liang J Tang	**Art Unit** 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *8,10-15 and 17-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *8,10-15 and 17-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to the application filed on 10/14/2004.

The priority date for this application is 07/31/2001.

Claims 1-7, 9 and 16 are cancelled. Claims 8, 10-15 and 17 are amended.

Claims 21-28 are added.

Claims 8, 10-15 and 17-28 are pending and have been examined.

Response to Arguments

2. Applicant's arguments with respect to claims 8, 10-15 and 17-28 have been considered but they are not persuasive.

Claims (21-24), (8, 10-12) and (28, 17-19) are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims (1-5), (8-11) and (18-19) of co-pending Application No. 09/886,454 (hereinafter '454) respectively. Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observation.

Claims 21-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 8, 10-12, 15, 17-24 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crelier, US Patent No. 6,151,703.

Claims 13-14 and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crelier in view Applicant's admitted prior art.

In the remarks, the applicant argues that:

As for independent claim 8, the Applicants argue that Crelier does not teach or suggest (E.g. see RE page 9, last paragraph):

(A) sequentially reading by the virtual machine at runtime the cluster of JAVA object representations,

(B) determining by the virtual machine at runtime whether JAVA objects or JAVA classes are to be identified,

(C) using said first references of said cluster to mark memory addresses that correspond to JAVA objects or JAVA classes, thereby allowing JAVA objects or JAVA classes to be identified at runtime by a sequential read of the cluster.

Examiner's response:

The examiner disagrees with Applicant's assertion.

In fact, Crelier does teaches

(A) sequentially reading (E.g. see col. 8:13-14 and col. 8:19-20) by said virtual machine at runtime (E.g. see col. 1:53) said cluster of JAVA object representations (E.g. see col. 3:44-45). Crelier does not explicitly disclose cluster. However, it would have been obvious to modify Crelier's teaching to expend from one array to more than one arrays (i.e. cluster). The modification would have been obvious because one of ordinary skill in the art would have been motivated to make the system more robust.

Furthermore, read the bytecode in object representation , the bytecode is inherently in order.

(B) determining by said virtual machine at runtime (E.g. see col. 1:53) whether JAVA objects (E.g. see col. 8:16-17) or JAVA classes (E.g. see col. 8:62-64) are to be identified (E.g. see col. 8:42-43, "thishash and totalhash").

(C) using said first references of said cluster to mark memory addresses that correspond to JAVA objects or JAVA classes, thereby allowing JAVA objects or JAVA classes to be identified at runtime (E.g. see col. 1:53) by a sequential read of the cluster (E.g. see col. 8:42, "unsigned long thishash").

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims (21-24), (8, 10-12) and (28, 17-19) are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims (1-5), (8-11) and (18-19) of co-pending Application No. 09/886,454 (hereinafter '454) respectively. Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observation.

Instant Claim	'454 Claim
21. In a JAVA computing environment, a virtual machine for identifying active	1. In a Java computing environment, a Java object representation suitable for use by a

JAVA objects and active JAVA classes at runtime, wherein said virtual machine is capable of:	Java virtual machine, said a Java object representation comprising:
generating and loading in the virtual machine prior to execution time a cluster of java object representations which are sequentially represented inside the virtual machine, wherein each of said JAVA object representations in said cluster consists of: a first reference to an internal class representation of a class associated with a JAVA object, and a second reference to instance fields associated with said JAVA object;	**a first reference to an internal class representation of said Java object;** **a second reference to instance fields associated with said Java object;** and wherein
sequentially reading by said virtual machine at runtime said cluster of JAVA object representations;	
determining by said virtual machine at runtime whether JAVA objects or JAVA classes are to be identified;	**said first reference is a direct reference to said internal class representation of said Java object.**

using said first references of said cluster to mark memory addresses that correspond to JAVA objects when said determining determines that JAVA objects are to be identified, thereby allowing JAVA objects to be identified at run time by a sequential read of said cluster; and using one or more of second first references of said cluster to mark memory addresses that correspond to JAVA classes when said determining determines that JAVA classes are to be identified, thereby allowing JAVA classes to be identified at run time by a sequential read of said cluster 22. A virtual machine as recited in claim 21, wherein **said first reference is a direct reference to said internal class representation of said Java object.** 23. A virtual machine as recited in claim	2. A Java object representation as recited in claim 1, wherein **said second reference is a reference to an array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.** 3. A Java object representation as recited in claim 1, wherein **said first reference is allocated as four bytes.** 4. A Java object representation as recited in claim 1, wherein **said second reference is allocated as four bytes.** 5. A Java object representation as recited in claim 1, wherein **said internal class representation**

21, wherein **said second reference is a reference to an array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.** 24. A virtual machine as recited in claim 21, wherein **said first and second references are allocated as four bytes.**	**includes a header of a predetermined size, and wherein a method table associated with said Java object is allocated immediately after said header.** .
8. In a JAVA computing environment, a method of identifying active JAVA objects and active JAVA classes by a virtual machine at runtime, said method comprising: generating and loading in the virtual machine prior to execution time a cluster of java object representations which are sequentially represented inside the virtual machine, wherein each of said JAVA object representations in said cluster consists of:	8. A method for representing a Java object in a virtual machine, said method comprising:

a first reference to an internal class representation of a class associated with a JAVA object, and a second reference to instance fields associated with said JAVA object;	
sequentially reading by said virtual machine at runtime said cluster of JAVA object representations;	
determining by said virtual machine at runtime whether JAVA objects or JAVA classes are to be identified;	allocating a first reference in a memory portion of said virtual machine, wherein **said first reference is a reference to an internal class representation of said Java object;**
using said first references of said cluster to mark memory addresses that correspond to JAVA objects when said determining determines that JAVA objects are to be identified, thereby allowing JAVA objects to be identified at run time by a sequential read of said cluster; and	allocating a second reference in a memory portion of said virtual machine, wherein **said second reference is a reference to instance fields associated with said Java object; and**
using one or more of second first references of said cluster to mark memory addresses that correspond to JAVA classes	

when said determining determines that JAVA classes are to be identified, thereby allowing JAVA classes to be identified at run time by a sequential read of said cluster	wherein **said first reference is a direct reference to said internal class representation of said Java object.**
10. A method as recited in claim 9, wherein **said first reference is a direct reference to said internal class representation of said Java object.**	9. A method as recited in claim 8, wherein **said second reference is a reference to an array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.**
11. A method as recited in claim 9, wherein **said second reference is a reference to an array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.**	10. A method as recited in claim 9, wherein **said first reference is allocated as four bytes.** 11. A method as recited in claim 9, wherein **said second reference is allocated as four bytes.**

12. A method as recited in claim 9, wherein **said first and second references are allocated as four bytes.**	
28. A computer readable medium including In a JAVA computing environment, a method of identifying active JAVA objects and active JAVA classes by a virtual machine at runtime, said method comprising: computer program code for generating and loading in the virtual machine prior to execution time a cluster of java object representations which are sequentially represented inside the virtual machine, wherein each of said JAVA object representations in said cluster consists of: a first reference to an internal class representation of a class associated with a JAVA object, and a second reference to instance fields associated with said JAVA object;	18. A computer readable media including computer program code for a Java object representation suitable for use by a Java virtual machine, said computer readable media comprising:

computer program code for sequentially reading by said virtual machine at runtime said cluster of JAVA object representations; computer program code for determining by said virtual machine at runtime whether JAVA objects or JAVA classes are to be identified; computer program code for using said first references of said cluster to mark memory addresses that correspond to JAVA objects when said determining determines that JAVA objects are to be identified, thereby allowing JAVA objects to be identified at run time by a sequential read of said cluster; and computer program code for using one or more of second first references of said cluster to mark memory addresses that correspond to JAVA classes when said determining determines that JAVA classes	computer program code for **a first reference to an internal class representation of said Java object;** computer program code for **a second reference to instance fields associated with said Java object; and** wherein **said first reference is a direct reference to said internal class representation of said Java object.** 19. A computer readable media as recited in claim 18, wherein **said second reference is a reference to an**

are to be identified, thereby allowing JAVA classes to be identified at run time by a sequential read of said cluster 17. A computer readable medium as recited in claim 28, wherein each of said Java object representations consists of: **a first reference to an internal class representation of a class associated with a Java object,** and **a second reference to instance fields associated with said Java object.** 18. A computer readable medium as recited in claim 17, wherein **said first reference is a direct reference to said internal class representation of said Java object.** 19. A computer readable medium as recited in claim 18, wherein said second reference is a reference to an	**array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.**

array of references, and wherein each reference in said array of references is a reference to an instance field associated with said Java object.	

The limitations recited in claims 21-24 are obvious variations of limitation in '454 Claims 1-5.

The limitations recited in claims 8 and 10-12 are obvious variations of limitation in '454 Claim 8-11.

The limitations recited in claim 28 and 17-19 is obvious variations of limitation in '454 Claim 18-19.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Specification

4. The abstract of the disclosure is objected to because "Java" is a trademark. Correction is required. See MPEP § 608.01(b).

The disclosure is objected to because of the following informalities: "Java" is a trademark. Appropriate correction is required.

Claims 8, 11, 14-15, 17-18, 21-23 and 26-28 objected to because of the following informalities: "Java" is a trademark. Appropriate correction is required.

The examiner suggest using language such as "bytecode programming language", "Java^{TM}" or "JAVA" to replace "Jave".

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 21-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 21 merely claims a virtual machine for identifying active JAVA objects and active JAVA classes at runtime. The virtual machine for identifying active JAVA objects and active JAVA classes at runtime are merely software components (e.g. computer program per se). Such claimed matter is descriptive material per se, non-functional descriptive material, and is not statutory because it is not a physical "thing" nor a statutory process, as there are no "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed aspects of the invention which permit the computer program's functionality to be realized. Since a computer program is merely a set of instructions capable of being executed by a computer, the program itself is not a process, without the computer-readable medium needed to realize the computer program's functionality. In contrast, a claimed computer-readable medium encoded with a computer program defines structural and functional interrelationships between the computer program and the medium which permit the computer program's functionality to be realized, and is thus statutory. **Warmerdam**, 33 F.3d at 1361, 31 USPQ2d at 1760. **In re Sarkar**, 558 F.2d 1330, 1333, 200 USPQ 132, 137 (CCPA 1978). See MPEP § 2106(IV)(B)(1)(a).

Claims 22-27, which depend from claim 21 are also rejected under 35 U.S.C. 101 for the same reason.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 8, 10-12, 15, 17-24 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crelier, US Patent No. 6,151,703.

As Per claim 8, Crelier teaches:

"generating and loading in the virtual machine prior to execution time a cluster of java object representations (E.g. see col. 3:44-45) which are sequentially represented inside the virtual machine, wherein each of said JAVA object representations in said cluster consists of:

> a first reference to an internal class representation of a class associated with a JAVA object(E.g. see FIG. 4, object handle 401 and associated text, e.g. col. 8:16-17), and

a second reference to instance fields associated with said JAVA object (E.g. see FIG. 4, pointer 421 and associated text, e.g. col. 8:21-22)";

"sequentially reading (E.g. see col. 8:13-14 and col. 8:19-20) by said virtual machine at runtime (E.g. see col. 1:53) said cluster of JAVA object representations (E.g. see col. 3:44-45)";

"determining by said virtual machine at runtime whether JAVA objects (E.g. see col. 8:16-17) or JAVA classes (E.g. see col. 8:62-64) are to be identified (E.g. see col. 8:42-43, "thishash and totalhash")";

"using said first references of said cluster to mark memory addresses that correspond to JAVA objects when said determining determines that JAVA objects are to be identified, thereby allowing JAVA objects to be identified at run time by a sequential read of said cluster" (E.g. see col. 8:42, "unsigned long thishash"); and

"using one or more of second first references of said cluster to mark memory addresses that correspond to JAVA classes when said determining determines that JAVA clases are to be identified, thereby allowing JAVA classes to be identified at run time by a sequential read of said cluster" (E.g. see col. 8:43, unsigned long totalhash")

Crelier does not explicitly disclose cluster. However, it would have been obvious to modify Crelier's teaching to expend from one array to more than one arrays (i.e. cluster). The modification would have been obvious because one of ordinary skill in the art would have been motivated to make the system more robust.

As Per claim 10, the rejection of claim 8 is incorporated and further Crelier teaches:

"wherein said first reference is a direct reference to said internal class representation of said Java object." (E.g. see col. 8:62-64).

As Per claim 11, the rejection of claim 10 is incorporated and further Crelier teaches:

"wherein said second reference is a reference to an array of references (E.g. see col. 8:13-14), and wherein each reference in said array of references is a reference to an instance field associated with said Java object (E.g. see col. 8:19-20)."

As Per claim 12, the rejection of claim 10 is incorporated and further Crelier teaches:

"wherein each of said first and second references are allocated in four bytes." (E.g. see col. 10:62-64).

As Per claim 15, the rejection of claim 10 is incorporated and further Crelier teaches:

"wherein said method is used by a virtual machine for garbage collection of Java objects and Java classes." (E.g. see col. 1:36-40).

As Per Claim 21, is the virtual machine claim corresponding to the method claim 8 and is rejected under the same reason set forth in connection of the rejection of claim 8.

As per Claims 22-24 and 27, the rejection of claim 21 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 10-12 and 15 respectfully.

As Per Claim 28, is the computer readable medium claim corresponding to the method claim 8 and is rejected under the same reason set forth in connection of the rejection of claim 8.

As Per claim 17, the rejection of claim 28 is incorporated and further Crelier teaches:

"a first reference to an internal class representation of a class associated with a Java object "(E.g. see FIG. 4, object handle 401 and associated text, e.g. col. 8:16-17), and

"a second reference to instance fields associated with said Java object" (E.g. see FIG. 4, pointer 421 and associated text, e.g. col. 8:21-22).

As per Claims 18-20, the rejection of claim 17 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 10-12 respectfully.

9. Claims 13-14 and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crelier in view Applicant's admitted prior art.

As Per claim 13, the rejection of claim 10 is incorporated and further Crelier does not explicitly disclose removing internal class representations that have not been marked. However, Applicant's admitted prior art teaches "removing internal class representations that have not been marked". (E.g. see Specification page 4, section 0011). Therefore, it would have been obvious to incorporate the teaching of Applicant's admitted prior art into the teaching of Crelier incorporate the method of removing internal class representations that have not been marked. The modification would have been obvious because one of ordinary skill in the art would have been motivated to perform a garbage collection to save resources.

As Per claim 14, the rejection of claim 10 is incorporated and further Crelier does not explicitly disclose removing Java objects that have not been marked. However, Applicant's admitted prior art teaches "removing Java objects that have not been marked". (E.g. see Specification page 4, section 0011). Therefore, it would have been obvious to incorporate the teaching of Applicant's admitted prior art into the teaching of Crelier incorporate the method of removing Java objects that have not been marked. The modification would have been obvious because one of ordinary skill in the art would have been motivated to perform a garbage collection to save resources.

As per Claims 25-26, the rejection of claim 21 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 13-14 respectfully.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuo-Liang J Tang whose telephone number is (571) 272-3705. The examiner can normally be reached on 8:30AM - 7:00PM (Monday – Thursday).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kuo-Liang J. Tang

Software Engineer Patent Examiner

JOHN CHAVIS
PATENT EXAMINER
ART UNIT 2124